INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:2229–2244
Published online 19 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1598

Improvements in speed for explicit, transient compressible flow solvers

Rainald Löhner[1,*,†], Hong Luo[2], Joseph D. Baum[3] and Darren Rice[4]

[1]*CFD Center, Department of Computational and Data Science, M.S. 6A2, College of Science, George Mason University, Fairfax, VA 22030-4444, U.S.A.*
[2]*Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695-7910, U.S.A.*
[3]*Advanced Concepts Business Unit, Science Applications International Corporation, McLean, VA, U.S.A.*
[4]*U.S. Government, Washington, DC, U.S.A.*

SUMMARY

Several explicit high-resolution schemes for transient compressible flows with moving shocks are combined in such a way so as to achieve the highest possible speed without compromising accuracy. The main algorithmic changes considered comprise the following:

- replacing limiting and approximate Riemann solvers by simpler schemes during the initial stages of Runge–Kutta solvers, and only using limiting and approximate Riemann solvers for the last stage;
- automatically switching to simpler schemes for smooth flow regions;
- automatic deactivation of quiescent regions; and
- unstructured grids with cartesian cores or embedded cartesian grids.

The results from several examples demonstrate that speedup factors of 1:4 are attainable without compromising the accuracy of the traditional FCT schemes. Copyright © 2007 John Wiley & Sons, Ltd.

Received 1 January 2007; Revised 5 July 2007; Accepted 6 July 2007

KEY WORDS: compressible flow solvers; explicit timestepping; computational techniques; TVD; FCT; CFD

1. INTRODUCTION

Many transient compressible flow problems are computed efficiently using explicit solvers. This is particularly the case for blast–structure interaction problems, where the fast propagation of shock

*Correspondence to: Rainald Löhner, CFD Center, Department of Computational and Data Science, M.S. 6A2, College of Science, George Mason University, Fairfax, VA 22030-4444, U.S.A.
†E-mail: rlohner@gmu.edu

Contract/grant sponsor: Defense Threat Reduction Agency (DTRA)

waves dominates the physics. Over the last two decades, the authors have developed and applied FCT and TVD solvers using unstructured grids to perform shock–object interaction simulations for complex geometries [1–8]. Of these, the FCT solvers seemed to offer the best accuracy/CPU performance. This paper summarizes the findings of a recent re-assessment of schemes. In view of some new ideas, such as automatic scheme switching, automatic deactivation and cartesian embedded grids, some interesting results appeared, which are the focus of this paper.

The Euler equations describing compressible flow as a system of conservation laws are given by

$$\mathbf{u}_{,t}+\nabla\cdot\mathbf{F}=0 \tag{1a}$$

$$\mathbf{u}=(\rho;\rho v_i;\rho e) \tag{1b}$$

$$\mathbf{F}_j=(\rho v_j;\rho v_i v_j+p\delta_{ij};v_j(\rho e+p)) \tag{1c}$$

where $\mathbf{u},\mathbf{F},\rho,v_i,e,p$ denote, respectively, the unknowns, fluxes, density, velocities, energy and pressure. Any finite volume or finite element discretization will yield a discrete system of the form

$$\mathbf{M}\cdot\mathbf{u}_{,t}=\mathbf{r}(\mathbf{u}) \tag{2}$$

or, in index notation

$$M^{ij}\hat{u}^j_{,t}=C^{ij}\mathscr{F}_{ij}=r^i \tag{3}$$

Here, $\mathbf{M},\hat{u}^j,C^{ij},\mathscr{F}^{ij}$ denote the mass matrix, vector of unknowns, edge coefficients for fluxes and edge fluxes, respectively. This system is integrated in time using an explicit k-step Runge–Kutta scheme of the form

$$\mathbf{M}(\mathbf{u}^{n+i}-\mathbf{u}^n)=\alpha_i\Delta t\,\mathbf{r}(\mathbf{u}^{n+i-1}) \tag{4}$$

The coefficients α_i are chosen according to desired properties, such as damping and temporal order of accuracy, e.g.

$$\alpha_i=\frac{1}{k-i+1} \tag{5}$$

2. TVD AND FCT SCHEMES

Let us first consider the traditional TVD approach. For the standard Galerkin approximation, we have

$$\mathscr{F}_{ij}=\mathbf{f}_i+\mathbf{f}_j \tag{6}$$

i.e. an equal weighting of fluxes at the end point of an edge. This (high-order) combination of fluxes is known to lead to an unstable discretization, and must be augmented by stabilizing terms to achieve a stable, low-order scheme. In what follows, we enumerate the most commonly used options in order to compare them.

DOI: 10.1002/fld

2.1. Limiting before flux evaluation

If we assume that the flow variables are constant in the vicinity of the edge endpoints i, j, a discontinuity will occur at the edge midpoint. The evolution in time of this local flowfield was first obtained analytically by Riemann [9], and consists of a shock, a contact discontinuity and an expansion wave. More importantly, the flux at the discontinuity remains constant in time. One can therefore replace the average flux of the Galerkin approximation by this so-called Riemann flux. This stable scheme, which uses the flux obtained from an exact Riemann solver, was first proposed by Godunov [10]. The flux is given by

$$\mathscr{F}_{ij} = 2f(\mathbf{u}_{ij}^{\mathrm{R}}) \tag{7}$$

where $\mathbf{u}_{ij}^{\mathrm{R}}$ is the local exact solution of the Riemann problem to the Euler equations, expressed as

$$\mathbf{u}_{lr}^{R} = \mathrm{Rie}(\mathbf{u}_l, \mathbf{u}_r) \tag{8}$$

where

$$\mathbf{u}_r = \mathbf{u}_i, \quad \mathbf{u}_l = \mathbf{u}_j \tag{9}$$

This scheme represents what one may call the 'ultimate first-order scheme'. All waves are taken into account, and the basic underlying physics are well reproduced. In order to achieve a higher-order scheme, the amount of inherent dissipation must be reduced. This implies reducing the magnitude of the difference $\mathbf{u}_i - \mathbf{u}_j$ by 'guessing' a smaller difference of the unknowns at the location where the Riemann flux is evaluated (i.e. the middle of the edge). The assumption is made that the function behaves smoothly in the vicinity of the edge. This allows the construction or 'reconstruction' of alternate values for the unknowns at the middle of the edge. The additional information required to achieve a scheme of higher order *via* these improved values at the middle of the edge can be obtained in a variety of ways:

- through continuation and interpolation from neighboring elements [11];
- *via* extension along the most aligned edge [12]; or
- by evaluation of gradients [13, 14].

The last option is the one most commonly used, but carries a considerable computational overhead: 15 gradients for the unknowns in 3-D can account for a large percentage of CPU time.

The inescapable fact stated in Godunov's theorem that no linear scheme of an order higher than one is free of oscillations implies that with these higher-order extensions, some form of limiting will be required. For a review of these, see [15]. It is important to note that this form of limiting is done *before flux evaluation*, and that, strictly speaking, it should be performed with characteristic variables. A typical Godunov-based scheme therefore has five main cost components:

- gradient-based reconstruction of higher-order approximations to the left and right states;
- transformation from conservative to characteristic variables;
- limiting;
- transformation from characteristic to conservative variables; and
- solution of the exact Riemann problem.

In the sequel, we will enumerate possible simplifications to each of these cost components, thereby deriving a whole spectrum of commonly used schemes.

DOI: 10.1002/fld

The solution of the (nonlinear) Riemann problem requires an iterative procedure which is expensive. Therefore, a considerable amount of effort has been devoted to obtain faster 'approximate Riemann solvers' that still retain as much of the physics as the basic Riemann problem [16–19]. These may be written abstractly as

$$\mathbf{u}_{lr}^{\mathrm{AR}} = A\,\mathrm{Rie}(\mathbf{u}_l, \mathbf{u}_r) \tag{10}$$

A widely used solver of this class is the one derived by Roe [17], given by

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\mathbf{A}^{ij}|(\mathbf{u}_i - \mathbf{u}_j) \tag{11}$$

where $|\mathbf{A}^{ij}|$ denotes the standard Roe matrix evaluated in the direction $\mathbf{l}_{ji} = \mathbf{x}_j - \mathbf{x}_i$, and $\mathbf{x}_i, \mathbf{x}_j$ are the coordinates of the end points i, j of the edge. Note that, as before, reducing the magnitude of the difference $\mathbf{u}_i - \mathbf{u}_j$ *via* reconstruction and limiting leads to schemes of higher order.

A further possible simplification can be made by replacing the Roe matrix by its spectral radius. This leads to a numerical flux function of the form

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}|(\mathbf{u}_i - \mathbf{u}_j) \tag{12}$$

where

$$|\lambda^{ij}| = |v_{ij}^k \cdot S_k^{ij}| + c^{ij} \tag{13}$$

and v_{ij}^k and c^{ij} denote edge values, computed as nodal averages, of the fluid velocity and speed of sound, respectively, and S_k^{ij} is the unit normal vector associated with the edge (i.e. in 3-D the unit normal of the finite volume surface associated with the edge). This can be considered as a centered difference scheme plus a second-order dissipation operator, leading to a first order, monotone scheme. Note that this scheme does not require any gradients, and, although of first order, is almost as inexpensive as the Galerkin/central scheme. In the sequel, we will denote this scheme as 'central/2'. A higher-order scheme can be obtained by a better approximation to the 'right' and 'left' states of the 'Riemann problem'. Given that for smooth problems through the use of limiters the second-order dissipation $|\mathbf{u}_i - \mathbf{u}_j|$ reverts to the fourth-order dissipation [20, 21], and that limiting requires a considerable number of operations, the next possible simplification is to replace the limiting procedure by a pressure sensor function. A scheme of this type may be written as

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda_{ij}|\left[\mathbf{u}_i - \mathbf{u}_j + \frac{\beta}{2}\mathbf{l}_{ji} \cdot (\nabla \mathbf{u}_i + \nabla \mathbf{u}_j)\right] \tag{14}$$

where $0 < \beta < 1$ denotes a pressure sensor function of the form [22]

$$\beta = 1 - \frac{|p_i - p_j + 0.5\mathbf{l}_{ji} \cdot (\nabla p_i + \nabla p_j)|}{|p_i - p_j| + |0.5\mathbf{l}_{ji} \cdot (\nabla p_i + \nabla p_j)|} \tag{15}$$

For $\beta = 0, 1$, second- and fourth-order damping operators are obtained, respectively. Several forms are possible for the sensor function β [23]. Although this discretization of the Euler fluxes looks like a blend of second- and fourth-order dissipation, it has no adjustable parameters. In the sequel, we will denote this scheme as 'central/2/4'. The scalar dissipation operator presented above still requires the evaluation of gradients. This can be quite costly for Euler simulations: for a typical multistage scheme, more than 40% of the CPU time is spent in gradient operations, even if a new

DOI: 10.1002/fld

dissipation operator is only required at every other stage. The reason lies in the very large number of gradients required: 15 for the unknowns in 3-D, and an additional three for the pressure. An alternative would be to simplify the combination of second- and fourth-order damping operators by writing out explicitly these operators:

$$d_2 = \lambda_{ij}(1-\beta)[\mathbf{u}_i - \mathbf{u}_j], \quad d_4 = \lambda_{ij}\beta\left[\mathbf{u}_i - \mathbf{u}_j + \frac{\mathbf{l}_{ji}}{2}\cdot(\nabla\mathbf{u}_i + \nabla\mathbf{u}_j)\right] \tag{16}$$

Performing a Taylor series expansion in the direction of the edge, we have

$$\mathbf{u}_i - \mathbf{u}_j + \frac{\mathbf{l}_{ji}}{2}\cdot(\nabla\mathbf{u}_i + \nabla\mathbf{u}_j) \approx \frac{\mathbf{l}_{ji}^2}{4}\left[\left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_j - \left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_i\right] \tag{17}$$

This suggests the following simplification, which neglects the off-diagonal terms of the tensor of second derivatives:

$$\frac{l^2}{4}\left[\left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_j - \left.\frac{\partial^2\mathbf{u}}{\partial l^2}\right|_i\right] \approx \frac{l^2}{4}[\nabla^2\mathbf{u}_j - \nabla^2\mathbf{u}_i] \tag{18}$$

and leads to the familiar blend of second- and fourth-order damping operators [24, 25]:

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda_{ij}|(1-\beta)[\mathbf{u}_i - \mathbf{u}_j] - |\lambda_{ij}|\beta\frac{l^2}{4}[\nabla^2\mathbf{u}_j - \nabla^2\mathbf{u}_i] \tag{19}$$

2.2. *Lax–Wendroff/Taylor–Galerkin*

The essential feature of Lax–Wendroff/Taylor–Galerkin schemes is the combination of time and space discretizations, leading to second-order accuracy in both time and space. An edge-based two-step Taylor–Galerkin scheme can readily be obtained by setting the numerical flux to

$$\mathscr{F}_{ij} = 2\mathbf{f}(\mathbf{u}_{ij}^{n+1/2}) \tag{20}$$

where

$$\mathbf{u}_{ij}^{n+1/2} = \frac{1}{2}(\mathbf{u}_i + \mathbf{u}_j) - \frac{\Delta t}{2}\left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij} \tag{21}$$

and $\partial\mathbf{f}^j/\partial x_j|_{ij}$ is computed on each edge and given by either

$$\left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij} \approx \frac{\mathbf{l}_{ij}}{\mathbf{l}_{ij}^2}\cdot(\mathbf{F}^i - \mathbf{F}^j), \quad \left.\frac{\partial\mathbf{f}^j}{\partial x_j}\right|_{ij} \approx \frac{\mathbf{D}_{ij}}{\mathbf{D}_{ij}^2}\cdot(\mathbf{F}^i - \mathbf{F}^j) \tag{22}$$

where $\mathbf{D}_{ij}$ denotes the edge coefficients for the advective terms obtained from the Galerkin approximation. The major advantage of this scheme lies in its speed, since there is no requirement of gradient computations, as well as limiting procedures for smooth flows. An explicit numerical dissipation (e.g. in the form of a Lapidus viscosity [26]) is needed to model flows with discontinuities. Taylor–Galerkin schemes by themselves are of little practical use for problems with strong shocks or other discontinuities. However, they provide high-order schemes with the best cost/performance ratio for the flux-corrected transport schemes presented below.

DOI: 10.1002/fld

2.3. Limiting after flux evaluation

Limiting after flux evaluation is the key idea inherent to all FCT schemes [1, 2, 27–29]. If we focus on high-order schemes of the Lax–Wendroff/Taylor–Galerkin family, the high-order increment may be written as

$$\mathbf{M}_{\mathrm{l}}\Delta\mathbf{u}^{\mathrm{h}}=\mathbf{r}+(\mathbf{M}_{\mathrm{l}}-\mathbf{M}_{\mathrm{c}})\Delta\mathbf{u}^{\mathrm{h}} \tag{23}$$

Here $\mathbf{M}_{\mathrm{l}}$ denotes the diagonal, lumped mass matrix, and $\mathbf{M}_{\mathrm{c}}$ the consistent finite element mass matrix. The low-order scheme is simply given by

$$\mathbf{M}_{\mathrm{l}}\Delta\mathbf{u}^{\mathrm{l}}=\mathbf{r}+c_{\mathrm{d}}(\mathbf{M}_{\mathrm{c}}-\mathbf{M}_{\mathrm{l}})\mathbf{u}^{n} \tag{24}$$

i.e. lumped mass matrix plus sufficient diffusion ($c_{\mathrm{d}}=O(1)$) to keep the solution monotonic. Subtracting these two equations yields the antidiffusive edge contributions

$$(\Delta\mathbf{u}^{\mathrm{h}}-\Delta\mathbf{u}^{\mathrm{l}})=\mathbf{M}_{\mathrm{l}}^{-1}(\mathbf{M}_{\mathrm{l}}-\mathbf{M}_{\mathrm{c}})(c_{\mathrm{d}}\mathbf{u}^{n}+\Delta\mathbf{u}^{\mathrm{h}}) \tag{25}$$

Note that no physical fluxes appear in the antidiffusive edge contributions. This may also be interpreted as advancing the physical fluxes with extra diffusion, thus assuring transport, conservation, etc. Thereafter, perform the antidiffusive step to enhance the solution as much as possible without violating monotonicity principles. The simplicity of the antidiffusive edge contributions for this class of scheme makes it both fast and very general, and has been one of the main reasons why this scheme has served the CFD community for more than 15 years without major alterations, in particular for the shock–object interaction problems considered here.

Table I summarizes the main ingredients of high-resolution schemes, indirectly comparing the cost of most current flow solvers.

Table I. Solvers and their algorithmic ingredients.

Solver	Riemann	Gradient	Char. Transf.	Limiting
Classic Godunov	Yes	Yes	Yes	Yes
Consvar Godunov	Yes	Yes	No	Yes
Consvar Roe/HLLC	Approx.	Yes	No	Yes
Central/2/lim	No	Yes	No	Yes
Central/2/4	No	Yes	No	No
Central/2/4/lap	No	No	No	No
Central/2	No	No	No	No
Central	No	No	No	No
Taylor–Galerkin	No	No	No	No
TG-FCT	No	No	No	Yes

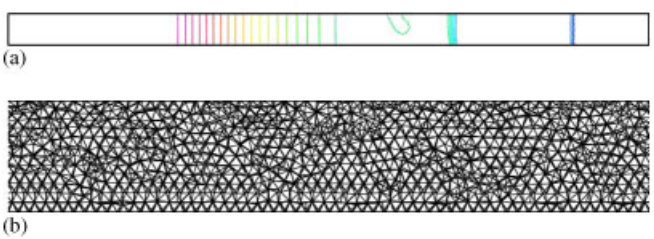

Figure 1. (a, b) Sod shock tube: density and mesh in plane.

DOI: 10.1002/fld

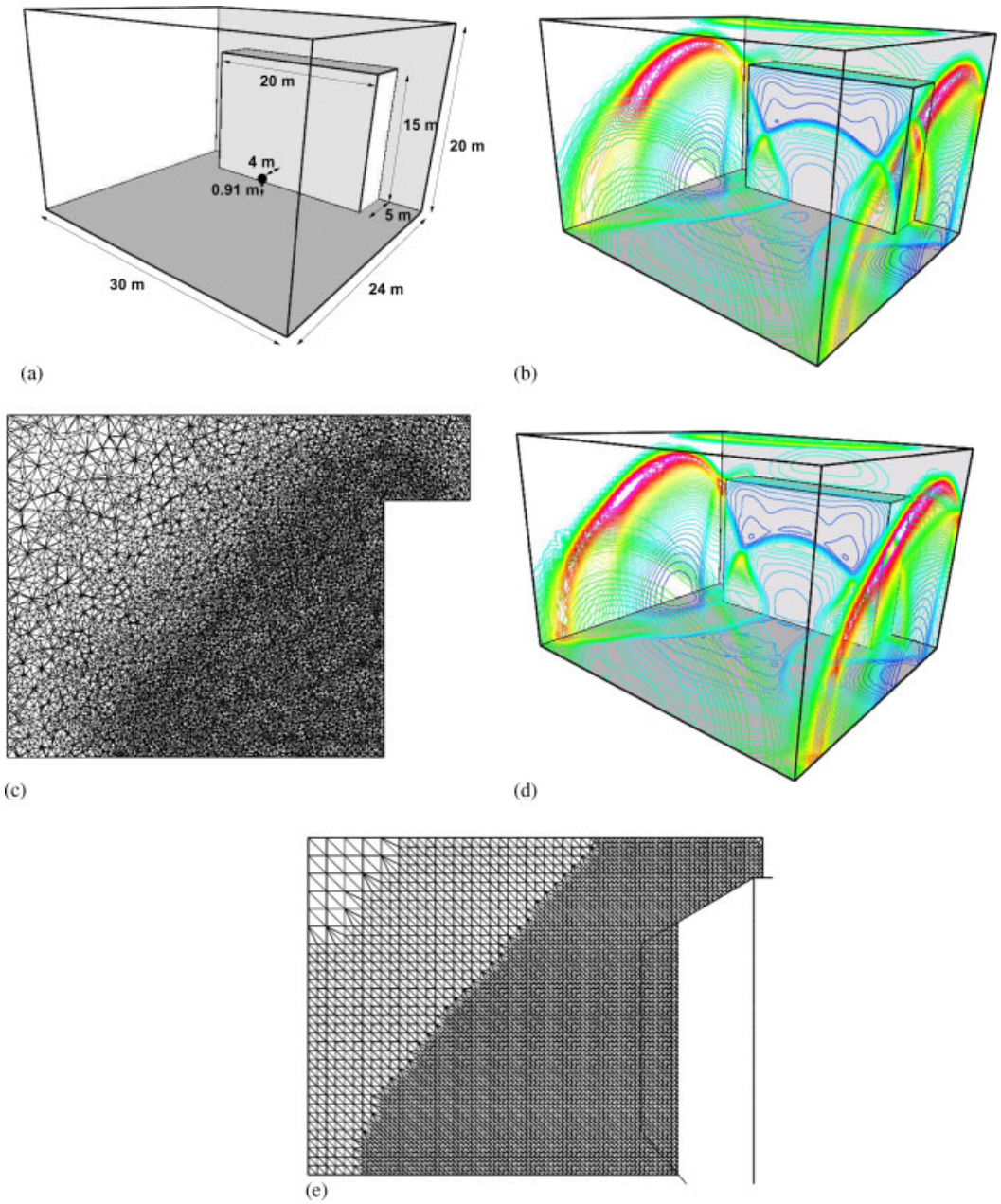

Figure 2. (a) Blast problem. (b,c) Blast problem: typical solution, mesh in symmetry plane. (d,e) Blast problem: embedded cartesian solution, mesh in symmetry plane.

DOI: 10.1002/fld

3. THE TEST CASES

We consider three cases. The first one is the well-known Sod shock tube problem [30], with initial states $(\rho, u, v, w, p)_1 = (1.0, 0.0, 0.0, 0.0, 1.0)$ and $(\rho, u, v, w, p)_1 = (0.1, 0.0, 0.0, 0.0, 0.1)$. Even though this case is a 1-D case, it is run in 3-D for comparison purposes.

The tube has a square cross-section, and the mesh chosen for comparison purposes had 34 Kpts, 173 Kels and 217 Kedgs. A typical solution for the density in plane through the center of the domain is shown in Figure 1(a). The corresponding triangulation, shown in Figure 1(b), gives an impression of the overall mesh used.

The second is a typical blast–structure interaction case. The geometry is depicted in Figure 2(a), and a typical solution in Figure 2(b). The initialization is taken from a detailed axisymmetric run that models the physics of TNT detonations. The TNT equivalent for this case is of 200 kg. This grid employed has 431 Kpts, 2468 Kels and 2924 Kedgs. A cut through the mesh in the plane of symmetry is shown in Figure 2(c). For comparison purposes, this case was also run with a mesh of 3.3 Mpts, 19.6 Mels and 23.05 Medgs. Figure 2(a) also shows the locations of eight stations where pressure time history data were recorded.

The third case is the same geometry and initialization as the second case, but run using the embedded option [31]. A cartesian mesh was generated for the box encompassing the domain. This mesh was adaptively refined to conform to the mesh size specifications of the previous case. A typical solution is shown in Figure 2(d), and a cut through the mesh in the plane of symmetry is shown in Figure 2(e). The cartesian nature of the mesh is clearly visible. This case has 462 Kpts, 2687 Kels and 3175 Kedgs.

All cases were run on a Dell 64-bit Intel-based PC with a clockspeed of 3.4 GHz and 4 Gb of RAM, using the Intel compiler under the Suse Linux OS. The only exception was the fine reference mesh, which as run on an SGI Altix with 32 Gb of RAM. Tables II and III list the subroutines that require the most amount of CPU resources for both the usual two-step Runge–Kutta HLLC approximate Riemann solver with van Albada limiting on conserved variables, as well as the usual FCT solver. These timings were taken for the second problem. Note that the FCT solver is marginally faster.

The first observation one can make by looking at Table II is that, as expected, the most expensive parts of the RK-HLLC solver are the limiting, approximate Riemann solver and gradient evaluation. This suggests replacing, as much as possible, this expensive solver by a cheaper solver. This can be done by modifying the the basic Runge–Kutta update, the flux evaluation, and by deactivating quiescent regions of the flow. These three concepts are treated in the following.

Table II. RK, two-stage, `nriem=4`, `nlimi=2`, $C=0.6$.

% Time	Seconds	Seconds	Name	Function
33.63	8466.40	8466.40	`limitra2`	van Albada limiter
21.30	13827.94	5361.54	`hllceualef`	HLLC approx. Riemann solver
10.79	16543.37	2715.43	`gradupd`	Gradient evaluation
7.50	18432.47	1889.10	`laploe1`	Lapidus art visc
Total	25175.98			

DOI: 10.1002/fld

Table III. FCT: `nlimi`$=6$ and $C=0.4$.

% Time	Seconds	Seconds	Name	Function
12.69	2623.15	2623.15	`gtadid`	Anti-diff flux
8.88	4459.08	1835.93	`tgeeuid`	Taylor–Galerkin fluxes
5.79	5654.64	1195.56	`laploed`	Lapidus art visc
5.36	6761.68	1107.03	`rhcomad`	Consistent mass matrix
4.96	7786.85	1025.17	`delupos`	Possible in/decrement
4.90	8799.53	1012.68	`rhspli`	rhs of limited fluxes
4.13	9652.05	852.52	`delupos`	Possible in/decrement
4.07	10493.58	841.53	`fluadta`	Anti-diff flux added/taken
Total	20664.53			

4. MODIFIED FLUXES FOR RUNGE–KUTTA STEPS

Given that any Runge–Kutta solver for first-order hyperbolic equations allows for larger Courant numbers with increasing stages, an interesting alternative is to evaluate stages $1:k-1$ in a k-stage scheme using inexpensive (albeit inaccurate) fluxes, and only employ the expensive, accurate flux evaluation for the last stage. Simple schemes that allow inexpensive flux evaluations are given by Equation (6) (central), Equation (12) (central/2) and Equations (20)–(22) (Taylor–Galerkin). Schemes of this kind have been used successfully in the past [16, 32, 33] for the class of shock propagation problems considered here. In the majority of cases two-stage Runge–Kutta schemes were considered (i.e. $k=2$). Using a purely central scheme (Equation (6)) for the first stage implies the risk of overshoots. Using the 'central/2' scheme (Equation (12)) for the first stage implies the risk of more dissipation in the solution.

5. MODIFIED FLUXES IN SMOOTH FLOW REGIONS

An observation made for many flowfields is that the regions of shocks and contact discontinuities only constitute a small fraction of the overall computational domain. It is in this relatively small region that the sophisticated, accurate and expensive schemes are required. In smooth flow regions, one could use less expensive second-order schemes such as those given by Equation (14). This idea has been used repeatedly, particularly for the more elaborate (and expensive) solvers for compressible transient flows [34, 35]. In a first step, the smoothness of the flow is determined. We have found the following indicator to be reliable for the class of problems considered here:

$$\xi = 1 - \max(\beta_1, \beta_2) \tag{26}$$

where

$$\beta_1 = \frac{|\rho_i - \rho_j + 0.5\mathbf{l}_{ji} \cdot (\nabla\rho_i + \nabla\rho_j)|}{|\rho_i - \rho_j| + |0.5\mathbf{l}_{ji} \cdot (\nabla\rho_i + \nabla\rho_j)| + c_{\mathrm{n}}(\rho_i + \rho_j)} \tag{27}$$

$$\beta_2 = \frac{|\mathbf{l}_{ji} \cdot (\nabla\rho_i - \nabla\rho_j)|}{|\mathbf{l}_{ji} \cdot \nabla\rho_i| + |\mathbf{l}_{ji} \cdot \nabla\rho_j| + c_{\mathrm{n}}(\rho_i + \rho_j)} \tag{28}$$

DOI: 10.1002/fld

Here c_n is a noise factor, which is typically taken as $c_n = O(0.01)$. For $\xi > 0.95$, the flow is considered smooth. Note that the gradients are available, as they are required for both the limited approximate Riemann solver, as well as the 'central/2/4 dissipation' (Equation (12)) fluxes. This also implies that there is little gain in going to the Laplacian form of the 'central/2/4 dissipation' (Equation (19)), which is less precise: the main difference in cost between these two options is the calculations of the gradients, a cost that has already been incurred.

6. DEACTIVATION IN QUIESCENT REGIONS

For many blast problems, many points remain in quiescent regions during a considerable portion of the run. This is especially true for point blasts. At the beginning, when high pressures and densities are present, the time steps are accordingly very small. This implies that during many time steps, regions away from the blast origin do not need to be updated. As the code processes groups of edges renumbered to avoid memory contention and avoid dirty cache lines [21], before each timestep all edge groups with all edges in quiescent regions are switched off. From this set of edges, the points that should not be updated are also marked. Quiescent regions are detected by evaluating the edge differences of density, momenta and energy. If any of these exceeds a preset tolerance, the edge is marked as active.

7. CARTESIAN GRIDS

For embedded CSD applications, an interesting option is to generate a regular, cartesian grid for the overall domain. This mesh is then subdivided into tetrahedra that are subsequently adaptively refined in order to conform to the specified spatial element size distribution. One can then proceed to remove the diagonal edges in most of the domain, ending up with schemes that resemble the usual cartesian finite difference methods, but also allow body-conforming elements close to boundaries, as well as adaptive mesh refinement within an unstructured solver [36]. The gains in CPU performance achievable by this approach have been shown to be considerable, and stem from several sources:

(a) The number of elements is typically smaller, as one considers as element size the side of a cubic cell.
(b) The removal of diagonal edges almost halves the number of edges that need to be processed.
(c) As the mesh is composed of nearly perfect elements, the allowable timestep is typically larger than for an equivalent unstructured mesh.
(d) Using bin-based point and edge renumbering, a large percentage of edges can be processed either without any or with partial indirect addressing.

8. RESULTS AND TIMINGS

8.1. Shock tube (Problem 1)

The 1-D plots obtained at the centerline of the domain using the different schemes are compared in Figure 3. Note that for this reference problem, all schemes yield very similar results.

DOI: 10.1002/fld

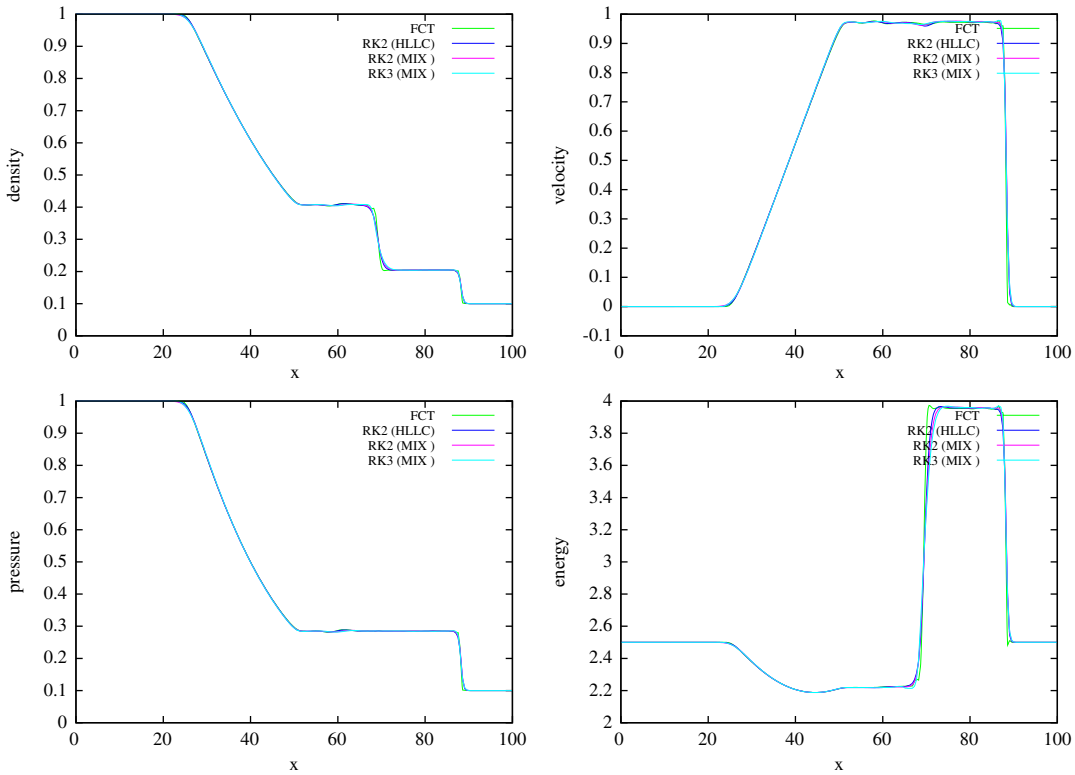

Figure 3. Shock tube problem: comparison of line-dumps.

Table IV. Timings for problem 1 (34 Kpts, 217 Kedgs).

Scheme	nstag	scaldi	switch	deac	coura	ntime	CPU
FCT	1	OFF	OFF	OFF	0.4	469	178
FCT	1	OFF	OFF	ON	0.4	469	134
RK	2	OFF	OFF	OFF	0.6	312	184
RK	2	OFF	OFF	ON	0.6	312	100
RK	2	ON	ON	OFF	0.6	312	84
RK	2	ON	ON	ON	0.6	312	55
RK	3	ON	ON	ON	1.2	236	38

The FCT results are a bit sharper than the equivalent RK/HLLC results. The difference between the usual (expensive) RK2/HLLC and the modified RK2/3 schemes (i.e. low-order central/2 for stages $1:k-1$, scheme switching based on local flow properties for stage k) is barely noticeable. The timings for the different options have been compiled in Table IV. In this and subsequent tables the following abbreviations have been employed: `nstag` denotes the number of stages (`nstag`$=1$ for Taylor–Galerkin/FCT, `nstag`>1 for RK/TVD), `scaldi` the activation of simpler schemes for the $k-1$ initial stages of a k-stage RK scheme, `switch` the switch to simpler schemes for

DOI: 10.1002/fld

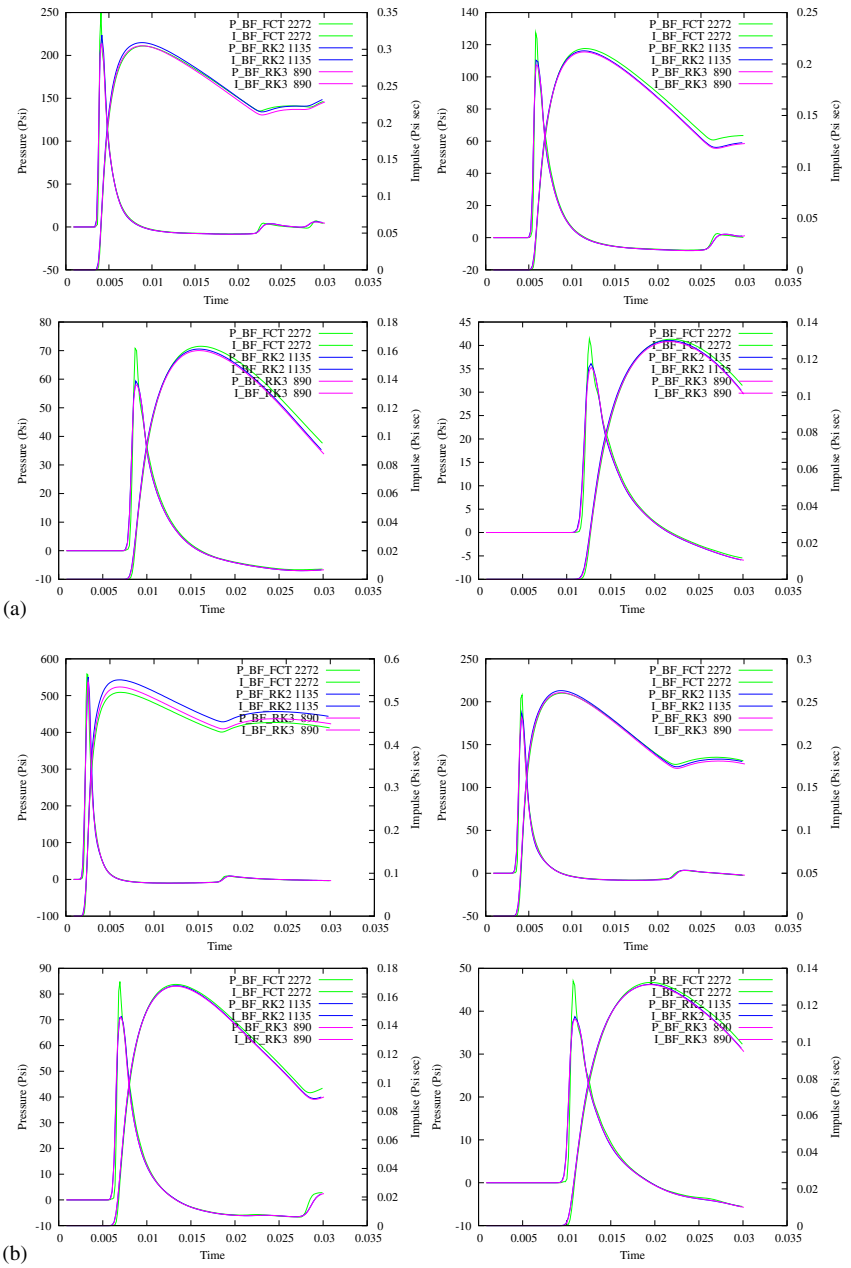

Figure 4. Blast problem: station time histories for (a) body fitted solutions (1–4) and (b) body fitted solutions (5–8).

DOI: 10.1002/fld

Table V. Timings for Problem 2 (430 Kpts, 2923 Kedgs).

Scheme	nstag	scaldi	switch	deac	coura	ntime	CPU
FCT	1	OFF	OFF	OFF	0.4	698	2272
FCT	1	OFF	OFF	ON	0.4	698	1995
RK	2	OFF	OFF	OFF	0.6	472	2463
RK	2	OFF	OFF	ON	0.6	472	2043
RK	2	OFF	ON	ON	0.6	471	1726
RK	2	ON	ON	ON	0.6	471	1135
RK	3	ON	ON	ON	1.2	236	890

the smooth regions of the flow, deac the deactivation of any update in the quiescent regions of the flow, coura the Courant number used, ntime the number of timesteps required to reach the physical time specified by the user and CPU the time required to complete the run. By the very nature of the physics and the resulting flowfields (with large deactive regions and large regions of constant flow), this first example shows considerable gains for all the options outlined above.

8.2. Blast on wall, body fitted (Problem 2)

The station time history data obtained for eight stations on the front wall using the different schemes are compared in Figure 4. Stations 1–8 (see Figure 2(a)) go from left to right, top to bottom, i.e. stations 1, 2 in the top row, then stations 3, 4 below, etc. The caption is coded as follows: (P,I) stands for pressure or impulse, BF for body fitted, FCT or RKn denote FCT or n-step Runge–Kutta schemes, and the number that follows denotes the time in seconds required for the run. The 'superfine' mesh solution (SFIN) is provided as a reference. Note that the FCT results give higher, sharper pressure peaks, but that for most stations the impulse (which is often used to assess wall damage) is very similar for all schemes. The times recorded have been listed in Table V. Deactivation yields a modest 10% gain in speed. However, in comparison with the original FCT scheme, the three-stage Runge–Kutta scheme with central/2 fluxes for the first two stages, automatic switch from HLLC to central/2/4, as well as automatic deactivation is approximately 2.5 times faster.

8.3. Blast on wall, embedded (Problem 3)

The station time history data obtained for eight stations on the front wall using the different schemes are compared in Figure 5. For reference, we include for each station the results obtained for the body-fitted approach on a mesh of similar size. One can notice that, as before, the FCT results give higher pressure peaks, but for most stations the impulse is very similar for all schemes. The times recorded have been listed in Table VI. Although the overall mesh has 3175 Kedgs, the number of active edges is only 1415 Kedgs, which accounts for the speedup of 2:1 as compared with the previous, unstructured, body-fitted case. Also note that in comparison with the original FCT scheme, the two-stage Runge–Kutta scheme with central/2 fluxes for the first stage, automatic switch from HLLC to central/2/4, as well as automatic deactivation is approximately 2.0 times faster. Overall, this scheme is approximately 4.0 times faster than the original FCT scheme for a body-fitted unstructured grid.

DOI: 10.1002/fld

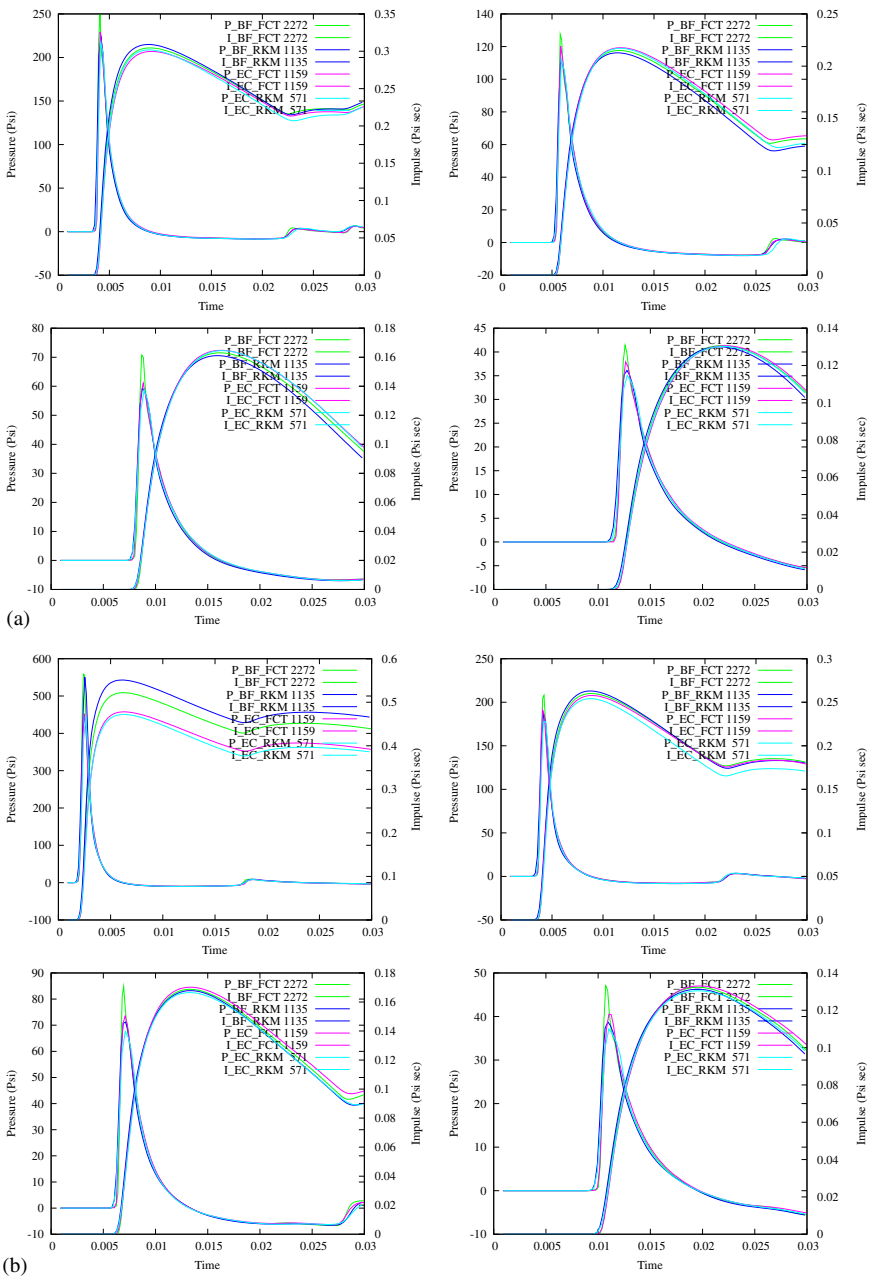

Figure 5. Blast problem: station time histories for (a) embedded solutions (1–4) and (b) embedded solutions (5–8).

DOI: 10.1002/fld

Table VI. Timings for Problem 3 (461 Kpts, 3174 Kedgs).

Scheme	nstag	scaldi	switch	deac	coura	ntime	CPU
FCT	1	OFF	OFF	OFF	0.4	490	1159
FCT+deac	1	OFF	OFF	ON	0.4	490	1064
RK 4/2/2	2	OFF	OFF	OFF	0.6	344	1046
RK 4/2/2+deac	2	OFF	OFF	ON	0.6	344	808
RK 24/12/2	2	ON	ON	OFF	0.6	355	648
RK 24/12/2	2	ON	ON	ON	0.6	347	571

9. CONCLUSIONS AND OUTLOOK

Several explicit high-resolution schemes for transient compressible flow have been combined in such a way so as to achieve the highest possible speed without compromising accuracy. The main algorithmic changes comprise the following:

- replacing limiting and approximate Riemann solvers by simpler schemes during the initial stages of Runge–Kutta solvers, and only using limiting and approximate Riemann solvers for the last stage;
- automatically switching to simpler schemes for smooth flow regions;
- automatic deactivation of quiescent regions; and
- unstructured grids with cartesian cores or embedded cartesian grids.

The results shown, as well as others obtained for the class of shock–object interaction problems considered here, demonstrate that speedup factors of 1:4 are attainable without significantly compromising the accuracy of the traditional FCT schemes. Thus, they pose an attractive alternative for shock–structure interaction calculations where *impulses* are sought. If, on the other hand, *peak pressures* are also required, the traditional FCT schemes offer a better quality result. Future work will center on improving limiting for the TVD schemes employed, in order to circumvent this shortcoming.

ACKNOWLEDGEMENTS

This research was partially supported by the Defense Threat Reduction Agency (DTRA). Drs Ali Amini and Douglas Sunshine acted as technical monitors.

REFERENCES

1. Löhner R, Morgan K, Peraire J, Vahdati M. Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1987; **7**:1093–1109.
2. Löhner R, Morgan K, Vahdati M, Boris JP, Book DL. FEM-FCT: combining unstructured grids with high resolution. *Communications in Applied Numerical Methods* 1988; **4**:717–730.
3. Baum JD, Luo H, Löhner R. Numerical simulation of a blast inside a Boeing 747. *AIAA-93-3091*, 1993.
4. Baum JD, Luo H, Löhner R. Numerical simulation of blast in the World Trade Center. *AIAA-95-0085*, 1995.
5. Baum JD, Luo H, Löhner R. The numerical simulation of strongly unsteady flows with hundreds of moving bodies. *AIAA-98-0788*, 1998.
6. Löhner R, Cebral J, Yang C, Baum JD, Mestreau E, Charman C, Pelessone D. Large-scale fluid–structure interaction simulations. *Computing in Science and Engineering* (CiSE) 2004; **May**/**June**:27–37.

DOI: 10.1002/fld

7. Baum JD, Mestreau E, Luo H, Löhner R, Pelessone D, Giltrud ME, Gran JK. Modeling of near-field blast wave evolution. *AIAA-06-0191*, 2006.
8. Luo H, Baum JD, Löhner R. A hybrid Cartesian grid and gridless method for compressible flows. *Journal of Computational Physics* 2006; **214**:618–632.
9. Riemann GFB. Über die Fortpflanzung ebener Luftwellen von Endlicher Schwingungweite. *Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen*, vol. 8, 1860.
10. Godunov SK. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik* 1959; **47**:271–306.
11. Billey V, Périaux J, Perrier P, Stoufflet B. 2-D and 3-D Euler computations with finite element methods in aerodynamic. *Proceedings of the International Conference on Hypersonic Problems*, Saint-Etienne, 13–17 January 1986.
12. Weatherill NP, Hassan O, Marchant MJ, Marcum DL. Adaptive inviscid flow solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes. *AIAA-93-3390*, 1993.
13. Whitaker DL, Grossman B, Löhner R. Two-dimensional Euler computations on a triangular mesh using an upwind, finite-volume scheme. *AIAA-89-0365*, 1989.
14. Luo H, Baum JD, Löhner R. Edge-based finite element scheme for the Euler equations. *AIAA Journal* 1994; **32**(6):1183–1190.
15. Sweby PK. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 1984; **21**:995–1011.
16. van Leer B. Towards the ultimate conservative scheme. II. Monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics* 1974; **14**:361–370.
17. Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
18. Osher S, Solomon F. Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics of Computation* 1982; **38**:339–374.
19. Luo H, Baum JD, Löhner R. Extension of Harten–Lax–van Leer scheme for flows at all speeds. *AIAA Journal* 2005; **43**(6):1160–1166.
20. Hirsch C. *Numerical Computation of Internal and External Flow*. Wiley: New York, 1991.
21. Löhner R. *Applied CFD Techniques*. Wiley: New York, 2001.
22. Peraire J, Peiro J, Morgan K. A three-dimensional finite element multigrid solver for the Euler equations. *AIAA-92-0449*, 1992.
23. Mestreau E, Löhner R, Aita S. TGV tunnel-entry simulations using a finite element code with automatic remeshing. *AIAA-93-0890*, 1993.
24. Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. *AIAA-81-1259*, 1981.
25. Mavriplis D. Three-dimensional unstructured multigrid for the Euler equations. *AIAA-91-1549-CP*, 1991.
26. Löhner R, Morgan K, Peraire J. A simple extension to multidimensional problems of the artificial viscosity due to Lapidus. *Communications in Applied Numerical Methods* 1985; **1**:141–147.
27. Kuzmin D, Turek S. Flux correction tools for finite elements. *Journal of Computational Physics* 2002; **175**: 525–558.
28. Kuzmin D, Möller M, Turek S. Multidimensional FEM-FCT schemes for arbitrary time-stepping. *International Journal for Numerical Methods in Fluids* 2003; **42**:265–295.
29. Zalesak ST. Fully multidimensional flux-corrected transport algorithm for fluids. *Journal of Computational Physics* 1979; **31**:335–362.
30. Sod G. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics* 1978; **27**:1–31.
31. Löhner R, Baum JD, Mestreau E, Sharov D, Charman C, Pelessone D. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering* 2004; **60**:641–660.
32. Colella P. Multidimensional upwind methods for hyperbolic conservation laws. *LBL-17023*, Preprint, 1983.
33. Colella P, Woodward P. The piecewise-parabolic method for hydrodynamics. *Journal of Computational Physics* 1984; **54**:174.
34. Fryxell B, Menon S. Large-Eddy simulation of Richtmeyer–Meshkov instability. *AIAA-05-0314*, 2005.
35. Liu C, Jiang L, Visbal M, Xie P. Smart weighted compact scheme for shock tube and shock–entropy interaction. *AIAA-06-3708*, 2006.
36. Löhner R, Luo H, Baum JD. Selective edge deactivation for unstructured grids with Cartesian cores. *Journal of Computational Physics* 2005; **206**(1):208–226.

DOI: 10.1002/fld